```
import os
import re
from email import message_from_file, message_from_string
from email.policy import default

# Adjust these patterns depending on your specific Mailman version / language
MAILMAN_PATTERNS = [
    re.compile(r"([\w\.-]+@[\w\.-]+\.\w+)\s+has\s+(?:been\s+)?removed",
re.IGNORECASE)
]

def extract_email_from_text(text):
    """Inspects email body text using common Mailman notification templates."""
    for pattern in MAILMAN_PATTERNS:
        match = pattern.search(text)
        if match:
            return match.group(1).strip()
    return None

def get_email_body(msg):
    """Safely extracts text/plain content from an email message object."""
    if msg.is_multipart():
        for part in msg.walk():
            content_type = part.get_content_type()
            content_disposition = str(part.get_content_disposition())
            if content_type == 'text/plain' and 'attachment' not in
content_disposition:
                try:
                    return part.get_content()
                except Exception:
                    return
part.get_payload(decode=True).decode(part.get_content_charset() or 'utf-8',
errors='ignore')
    else:
        try:
            return msg.get_content()
        except Exception:
            return msg.get_payload(decode=True).decode(msg.get_content_charset() or
'utf-8', errors='ignore')
    return ""

def process_eml_directory(directory_path, output_file):
    """Processes a directory of single .eml files."""
    unsubscribed_emails = set()

    if not os.path.exists(directory_path):
        print(f"Error: Directory '{directory_path}' does not exist.")
        return

    print(f"Scanning directory: {directory_path}")
```

```
    for filename in os.listdir(directory_path):
        if filename.endswith(".eml") or filename.endswith(".txt"):
            file_path = os.path.join(directory_path, filename)
            try:
                with open(file_path, 'r', encoding='utf-8', errors='ignore') as f:
                    msg = message_from_file(f, policy=default)
                    body = get_email_body(msg)
                    extracted = extract_email_from_text(body)
                    if extracted:
                        unsubscribed_emails.add(extracted.lower())
            except Exception as e:
                print(f"Failed parsing file {filename}: {e}")

    save_results(unsubscribed_emails, output_file)

def save_results(email_set, output_file):
    """Writes gathered unique email addresses sorted alphabetically to a file."""
    sorted_emails = sorted(list(email_set))
    with open(output_file, 'w', encoding='utf-8') as out:
        for email in sorted_emails:
            out.write(f"{email}\n")
    print(f"\nSuccess! Extracted {len(sorted_emails)} unique unsubscribed emails
into '{output_file}'.")

if __name__ == "__main__":
    # --- CONFIGURATION ---
    # Path to the directory holding your saved Mailman email files
    TARGET_DIR = "c:\\projects3\\namecheap\\salvaging-unsubscribed-people\\pct"
    OUTPUT_TXT = "unsubscribed_list_pct.txt"
    # ---------------------

    # Create sample folder if it doesn't exist yet for user onboarding
    if not os.path.exists(TARGET_DIR):
        os.makedirs(TARGET_DIR)
        print(f"Created '{TARGET_DIR}' folder. Please place your .eml files inside
it and run again.")
    else:
        process_eml_directory(TARGET_DIR, OUTPUT_TXT)
```